ESAIM: Proceedings
Actes du 32ème Congrès
d'Analyse Numérique : Canum2000
URL: http://www.emath.fr/proc/Vol.8/

Vol. 8, Septembre 2000, 1-3

Représentations non structurées en optimisation topologique de formes par algorithmes évolutionnaires

Hatem Hamda[1], François Jouve[1], Evelyne Lutton[2],
Marc Schoenauer[1] et Michèle Sebag[3]

Résumé. Les algorithmes évolutionnaires sont des méthodes d'optimisation stochastiques inspirées – grossièrement – de l'évolution naturelle des populations. Méthodes globales d'ordre 0, leur robustesse et leur souplesse leur permettent d'attaquer la résolution numérique de problèmes difficiles à résoudre autrement. Mais c'est leur capacité à travailler sur des espaces de recherche non standards qui leur offre les perspectives les plus originales.

Dans le domaine de l'optimisation topologique de formes, les résultats obtenus il y a quelques années montraient la faisabilité de l'approche évolutionnaire, mais étaient limités par le fait que la complexité de l'espace de recherche était liée à celle du maillage utilisé lors de la simulation numérique. Ce papier introduit un ensemble de représentations compactes et non structurées dont la complexité n'est pas fixe, mais est ajustée par l'algorithme lui-même. Les résultats présentés dans cet article montrent que leur utilisation permet de repousser les limites de l'optimisation topologique de formes évolutionnaire. Des résultats sur des problèmes test simples tentent ensuite de comparer les mérites des diverses approches proposées.

Mots clés. Optimisation topologique de formes, algorithmes évolutionnaires, représentations non structurées.

Abstract. *Evolutionary Algorithms are stochastic optimization methods based on a crude mimic of natural evolution. The robustness and flexibility of these zero-th order methods allows to overcome many limits of classical deterministic optimization algorithms. But the most significant breakthroughs can be achieved thanks to their ability to use non standard search space.*

This paper focuses on Topological Optimum Design problems : early results have shown the ability of Evolutionary methods to find numerical solutions to yet unsolved problems, but those approaches were limited because the complexity of the representation was that of a fixed underlying mesh. Different new representations, both unstructured and compact, are introduced : their complexity is self-adaptive, i.e. is evolved by the algorithm itself. This paper demonstrates that this allows one to push further the limits of Evolutionary Topological Optimum Design. Results on simple test problems then try to compare the different representations proposed here.

Key words. *Topological Optimum Design, Evolutionary Computation, unstructured representations.*

1 Introduction

Les algorithmes évolutionnaires (AE) [6] sont des algorithmes d'optimisation stochastique basés sur une imitation grossière de l'évolution Darwinienne des populations. Pour optimiser une fonction-objectif donnée $\mathcal{F}$ (appelée aussi *performance* ou *fitness*) définie sur un espace de recherche E, une *population* d'*individus* (points de E) est soumise à une suite de *générations* (la première génération est tirée au hasard dans E). Une génération commence par la *sélection* de quelques individus bien adaptés (par rapport à $\mathcal{F}$) pour la reproduction. Ces individus engendrent une *descendance* en utilisant des opérateurs stochastiques appelés *croisement* pour les opérateurs binaires, et *mutations* pour les opérateurs agissant su un seul individu. Enfin, quelques-uns des descendants *remplacent* certains des parents pour terminer le processus de génération. Les paradigmes de sélection et de remplacement, qui représentent les étapes de la règle Darwinienne de la *survie du mieux adapté*, peuvent être stochastiques ou déterministes. Comme dans l'évolution naturelle, on espère l'émergence progressive d'individus de mieux en mieux adaptés : les meilleurs individus de la population finale (au regard de $\mathcal{F}$) sont des approximations de solutions du problème d'optimisation posé.

Très souvent, l'espace de *représentation* sur lequel on fait effectivement la recherche (sur lequel les opérateurs d'évolution opèrent), appelé également l'espace des *génotypes*, est différent de l'espace dans lequel la performance est calculée, appelé l'espace des *phénotypes*. Par exemple, en optimisation de structures, si on cherche une forme optimale définie par sa frontière représentée par des splines s'appuyant des points de contrôle en nombre fixe, les génotypes sont des vecteurs réels de taille (finie) donnée. En revanche, l'espace des phénotypes est l'ensemble des formes dont le comportement mécanique sert à calculer la fonction-objectif.

Dans le contexte de l'optimisation paramétrique, i.e. lorsque l'espace de recherche est de dimension finie, les AE sont simplement une technique de plus, à la fois une puissante méthode d'ordre 0 (seules les valeurs de la fonction-objectif sont nécessaires), et une méthode globale stochastique capable de s'échapper de minima locaux pour trouver un optimum global, ou même des optima globaux multiples dans le cas de fonctions multimodales. Les AE ont été employés dans de nombreux domaines : pour des problèmes de combinatoire discrète (depuis la TSP standard [42] ou les problèmes de coloration de graphes [16] jusqu'à des problèmes réels d'attribution de fréquences [15] ou de scheduling [47]), aussi bien que des problèmes continus (depuis la célèbre expérience d'optimisation de la forme d'une tuyère [49, 56] jusqu'à de plus récentes applications industrielles [20, 59, 46]).

Toutefois, une particularité des AE est son aptitude à traiter des génotypes très inhabituels comme les espaces de graphes, de listes non ordonnées. . .La seule contrainte est de fournir une procédure d'initialisation et des opérateurs d'évolution qui respectent quelques propriétés [58]. En effet, plus l'espace de recherche est grand, meilleure seront les solutions mais elles seront d'autant plus difficiles à atteindre. Pourtant, les succès les plus spectaculaires des AE ont été obtenus en utilisant des représentations non structurées, c-à-d des représentations non paramétriques. Reprenons l'exemple des splines et considérons maintenant un nombre de nœuds variable (avec des positions variables également) : cela donne une représentation non structurée et il n'est pas très difficile d'imaginer des opérateurs d'évolution adaptés permettant de faire évoluer de telles représentations. Par exemple, la fameuse expérience pionnière de la tuyère par Rechenberg et Schwefel utilisait une représentation de longueur variable [49, 56].

Mais il y a une étape supplémentaire qui nous éloigne des représentations directes : dans les approches appelées *morphogénétiques*, au lieu de chercher une solution du problème de départ, l'idée consiste à chercher un "programme" qui, en s'exécutant, va construire une solution. La performance du programme étant celle de la solution qu'il permet de construire. Toute la Programmation Génétique (Genetic Programming) [34] appartient à cette catégorie de représentations, et des résultats impressionnants ont été obtenus en utilisant ces techniques, par exemple dans le domaine de la conception de circuits électroniques analogiques [35].

Dans le cadre de l'optimisation topologique de formes, ce papier s'intéresse aux problèmes de représentation pour les structures : la représentation paramétrique directe basée sur un maillage donné du domaine de travail et un tableau de bits, bien que très employée et utile pour une première comparaison avec les approches déterministes [32, 30, 33], montre ses limites lorsque l'on veut augmenter la complexité des maillages. Nous proposons ici différentes représentations non structurées basées sur les diagrammes de Voronoï, qui ont une complexité auto-adaptative (i.e. la complexité des solutions est ajustée par l'algorithme), mais qui nécessitent tout de même des "gènes" élémentaire définis par le programmeur. Enfin, nous introduisons la représentation IFS – une approche morphogénétique basée sur la théorie des fractales – qui est une tentative pour élargir l'espace de recherche de telle sorte qu'aucun *a priori* sur la structure de "gène" élémentaire n'est plus nécessaire. Des résultats originaux utilisant la représentation Voronoï confirment la puissance de l'approche non structurée. Des comparaisons sur le benchmark du cantilever permettent de choisir parmi les différentes représentation à base de diagrammes de Voronoï et de cerner les limites de l'approche morphogénétique, au moins pour des problèmes simples d'optimisation topologique de structures.

Cet article est organisé de la façon suivante : le contexte de l'optimisation topologique de structures par AE est rappelé dans la section 2, depuis les bases mécaniques jusqu'aux techniques de pénalisation adaptatives utilisées à l'intérieur de la fonction objectif. La section 3 introduit trois différentes représentations basées sur les diagrammes de Voronoï, tandis que la section 4 montre des résultats numériques originaux obtenus avec une représentation Voronoï simple, et des comparaisons entre les trois représentations de type Voronoï. Dans la section 5, on introduit encore une nouvelle représentation basée sur la théorie fractale des systèmes de fonctions itérées (IFS), ainsi que des résultats préliminaires montrant ses potentiels avantages. Enfin, la section 6 résume l'article et discute de la pertinence des différentes représentation non structurées compactes introduites auparavant.

2 Rappels

2.1 Le problème mécanique

Le contexte de ce papier est Optimisation Topologique de Structures (Topological Optimum Design – TOD). Le problème consiste à trouver la forme optimale d'une structure contenue dans un domaine donné (i.e. la répartition de matière dans ce domaine) de telle sorte que le comportement mécanique de cette structure satisfasse certaines contraintes – ici par exemple une borne sur le le déplacement maximal sous un chargement donné, mais on pourrait aussi imaginer une borne sur des fréquences propres ou une combinaison de critères mettant en jeu la rigidité et le comportement vibratoire. Le critère à optimiser est ici le poids de la structure mais on pourrait aussi introduire par exemple des coûts de fabrication.

Le modèle mécanique utilisé ici est celui de l'élasticité linéarisée bidimensionnelle en contraintes plane (à l'exception de la section 4.4 où les résultats sont tridimensionnels), et on ne considérera que des matériaux linéaires et isotropes (cf. e.g. [13]). Toutes les figures sont adimensionnelles (e.g. le module d'Young vaut toujours 1) et les effets de gravité sont négligés.

Un benchmark très populaire pour l'optimum design et celui de la plaque-console (cantilever) : le domaine de calcul est un rectangle, la plaque est encastrée sur la partie verticale gauche de la frontière (déplacement imposé à 0) et le chargement consiste à appliquer une force ponctuelle verticale au milieu de la frontière verticale de droite. La Figure 1 montre le domaine de calcul pour le cantilever 2×1.

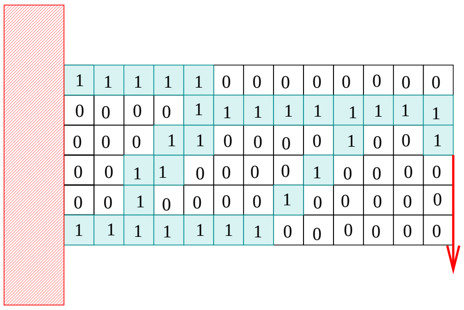


FIG. 1 – *Le problème de la plaque-console* 2×1 *et une représentation "bitarray" (tableau de bits) d'une structure sur un maillage régulier (ici un maillage* 13×6*). Cf. section 3.*

2.2 Rapide état de l'art en optimisation de formes

Les principales tendances en optimisation de structures sont les suivantes. La première approche se nomme *variation de domaine* [10] ou *analyse de sensibilité*. Elle consiste en des variations successives d'un domaine initial et est basée sur le calcul d'un gradient de la fonction-objectif par rapport aux variables définissant la forme. Cette approche possède deux défauts majeurs : elle nécessite une bonne intuition de la forme initiale et se montre instable pour de grandes variations du domaine ; de plus, elle ne permet pas de modifier la *topologie* de la forme initiale (e.g. ajouter ou supprimer des trous). Toutefois, l'introduction récente du gradient topologique [18] permet dans une certaine mesure de contourner cette dernière limitation, mais uniquement dans le cadre de l'élasticité linéaire.

Une autre méthode d'optimisation topologique est l'approche introduite en 1988 dans [7] et maintenant standard utilisant l'homogénéisation. La forme optimale est recherchée comme une densité de matière en chaque point (comprise entre 0 et 1) et une microstructure locale décrivant la forme du mélange matière/vide. L'introduction de telles microstructure dans l'espace de recherche revient à *relaxer* le problème initial (au sens mathématique du terme) pour obtenir un problème bien posé [2]. Les algorithmes numériques issus de telles formulations convergent vers des solutions généralisées (non physiques) faites de matériaux composites. Elles peuvent être post-traitées pour obtenir une solution admissible avec une densité booléenne [1]. Ces méthodes sont également jusqu'à présent limitées à l'élasticité linéaire. Les résultats théoriques sur les microstructure optimales ne concernent que le cas mono-chargement, bien que des résultats numériques aient été obtenus pour le cas multi-chargement [3]. De plus, cette méthode ne peut pas – encore – traiter le cas où les forces s'appliquent sur la frontière de la structure (qui est inconnue) comme par exemple le cas de la pression hydrostatique.

Une alternative possible à ces différentes techniques consiste à utiliser des méthodes d'optimisation stochastiques.

Les méthodes d'optimisation stochastiques ont été appliquées avec succès à d'autres problèmes d'optimisation de structures. Par exemple au cas des treillis pour le dimensionnement des sections des barres [40, 55] ou bien pour l'optimisation de la topologie du treillis [22, 9]. L'optimisation de matériaux composites a aussi été abordée par ce biais [36].

Des problèmes d'optimisation topologique de structures massives ont également été traités par des méthodes stochastiques : le recuit simulé a été utilisé pour optimiser la section d'une poutre [4] et des AE on permis de résoudre des problèmes de cantilever semblables à ceux présentés dans la section 2.1 (cf. [26, 12, 32]).

Quelques limitations des méthodes déterministes ont été levées dans ces travaux. Par exemple dans [32, 30, 33] des résultats en élasticité non linéaire ainsi que l'optimisation d'un dôme sous-marin (pour lequel le chargement s'applique sur une frontière inconnue) sont présentés. Ces deux cas ne peuvent pour le moment pas se traiter par les méthodes utilisant l'homogénéisation.

2.3 Calcul de la performance

Le problème traité dans cet article consiste à trouver une structure de poids minimal tel que son déplacement maximal – lorsqu'elle est soumise au chargement donné (cf. Figure 1) – reste inférieur à une valeur limite donnée D_{lim}. Le calcul du déplacement maximal est effectué en utilisant un programme classique d'éléments finis [28].

D'un point de vue mécanique, les structures qui ne relient pas le point d'application de la force et la frontière encastrée ne sont pas admissibles et on leur attribue une performance arbitrairement grande. De plus, toutes les parties de la structure qui ne sont pas connectées au point d'application de la force – et ainsi n'affectent pas le comportement mécanique de la pièce – sont éliminées avant l'analyse par éléments finis mais pénalisent la performance du fait de leur poids additionnel inutile (cf. [32, 30] pour une discussion détaillée de ces sujets). En résumé, pour les structures connexes, le problème est de minimiser le poids avec une contrainte du type $D^i_{Max} \leq D^i_{lim}$ pour chaque chargement, où D^i_{Max} est le déplacement maximal calculé par éléments finis pour le chargement i, et D^i_{lim} sa limite supérieure imposée. Pour tenir compte des contraintes, on utilise une méthode de pénalisation adaptative, que nous décrirons succinctement dans le cadre de l'optimisation topologique (cf. [8] pour une description détaillée dans un cadre plus général).

2.3.1 Méthodes de pénalisation

Soit α un *paramètre de pénalisation* positif (dans le cas d'une seule contrainte), la fonction-coût à minimiser est

$$Weight + \alpha(D_{max} - D_{lim})^+ \tag{1}$$

Toutefois, l'ajustement de α (qui peut être vu comme un multiplicateur de Lagrange pour la contrainte associée) n'est pas facile [43]. Une méthode de pénalisation statique, avec α constant, peut donner de très bons résultats mais nécessite un ajustement très fin de la valeur de α. La pénalisation dynamique, dans laquelle α est modifié suivant une stratégie spécifiée par l'utilisateur, comme il est proposé dans [27] ou bien, pour l'optimisation de formes dans [32], requière une bonne intuition initiale pour déterminer une stratégie efficace. L'utilisation d'une technique adaptative s'impose ici.

La plupart des algorithmes évolutionnaires utilisent de nombreux paramètres que l'utilisateur doit ajuster pour chaque problème particulier, et la technique systématique par essai-erreur est évidemment très coûteuse. Différentes méthodes, regroupées sous la terminologie générale de *techniques adaptatives* ont été proposées pour remédier à ce problème : les valeurs des paramètres peuvent être déduits de statistiques sur les itérations précédentes – comme dans la règle du 1/5ème pour la taille du pas de mutation proposée par [49] – ou bien évoluer suivant les opérateurs d'évolution [56]. Un petit nombre de paramètres doivent encore être ajustés par l'utilisateur (e.g. les valeurs initiales et les schémas de mise à jour), mais on peut les considérer comme des paramètres du second ordre : dans une large plage de valeurs, l'algorithme restera robuste. Pour une revue des techniques adaptatives dans les algorithmes évolutionnaires, cf. [25].

Des paramètres de pénalisation adaptatifs ont été utilisés avec succès pour des problèmes de satisfaction de contraintes discrètes (Constraint Satisfaction Problems) [17], où l'objectif est de trouver au moins un individu admissible. Dans le contexte de l'optimisation de paramètres, des schémas adaptatifs ont été proposés dans [23] et [57] : le paramètre de pénalisation est mis à jour en fonction de la faisabilité du meilleur individu dans la population des générations précédentes.

La méthode de pénalisation adaptative que nous avons utilisée effectue la mise à jour du paramètre de pénalisation en utilisant les statistiques globales de faisabilité dans la population. Le but de cette démarche est d'explorer les environs de la frontière de la région admissible en essayant de conserver dans la population les individus qui se situent "de part et d'autre" de cette frontière (cette même idée mène aux algorithmes génétiques ségrégués – Segregated GA – [37] qui utilisent deux différents paramètres de pénalisation pour atteindre ce but).

En effet, dans de nombreux problèmes d'optimisation, on sait que la solution se situe sur la frontière de la région admissible. Des techniques spécifiques de traitement des contraintes ont été proposées pour explorer uniquement cette frontière [53, 54].

Toutefois, pour l'optimisation topologique de formes, avec la rigidité comme fonction-objectif, la solution n'est pas *a priori* sur la frontière pour le problème continu. Une fois discrétisé, pour une taille de maillage fixe, ce n'est plus vrai et on peut dire que la solution est "proche" de la frontière, en un sens que nous ne préciserons pas ici. De plus, cette frontière est difficilement caractérisable pour ce problème, mais on peut explorer la zone admissible voisine de la frontière grâce à la méthode de pénalisation adaptative.

2.3.2 Pénalisation adaptative basée sur la population

Le but est de garder dans la population une proportion minimale d'individus satisfaisant les contraintes ainsi qu'une proportion minimale qui les violent. Notons $\Theta^k_{feasible}$ la proportion d'individus qui satisfont les contraintes à la génération k, et Θ_{inf} et Θ_{sup} deux paramètres donnés par l'utilisateur. Les faibles valeurs des paramètres de pénalisation favorisent les individus qui violent les contraintes (et inversement). Pour maintenir $\Theta^k_{feasible} \in [\Theta_{inf}, \Theta_{sup}]$, nous proposons la règle de mise à jour suivante :

$$\alpha_{k+1} = \begin{cases} \beta \cdot \alpha_k & \text{si } \Theta^k_{feasible} < \Theta_{inf} \\ (1/\beta) \cdot \alpha_k & \text{si } \Theta^k_{feasible} > \Theta_{sup} \\ \alpha_k & \text{sinon} \end{cases} \tag{2}$$

avec $\beta > 1$. Les paramètres choisis par l'utilisateur dans cette méthode sont Θ_{inf}, Θ_{sup}, β et la valeur initiale α_0. Après quelques tests de robustesse, les valeurs suivantes ont été choisies et seront utilisées dans toutes les simulations présentées dans cet article : $\beta = 1.1, \Theta_{inf} = 0.4$, et $\Theta_{sup} = 0.8$.

Notons que les variations de α ne sont pas monotones, et il n'y a donc pas de garantie *a priori* que le meilleur individu de la population satisfasse les contraintes. Il peut même arriver que la population ne contienne aucun individu admissible – même si dans ce cas, l'augmentation régulière de la valeur de α doive favoriser les individus violant le moins les contraintes, menant à l'émergence rapide d'individus admissibles.

Des résultats comparatifs montrant la validité de cette approche adaptative basée sur la population se trouvent dans [8] pour des problèmes modèles, et dans [24] pour des problèmes d'optimisation topologique.

2.4 Représentations des structures pour l'optimisation topologique de formes

Le point le plus crucial dans la construction d'un algorithme évolutionnaire est les choix de la représentation. Tous les travaux cités à la section 2.2 montrant des applications des AE à des problèmes d'optimisation topologique de formes utilisent la même représentation binaire "naturelle", appelée bitarray dans [32] : elle est associée à un maillage particulier du domaine – celui qui est utilisé pour calculer le comportement mécanique de la structure et déterminer la performance (cf. la section 2.3). A chaque élément du maillage on attribue une valeur 1 si il contient de la matière, et 0 sinon (cf. Figure 1). Notons que cette représentation binaire n'est pas équivalente à la représentation "bitstring" habituelle : un opérateur de croisement spécifique, utilisant la géométrie du problème, a été développé [31] qui est similaire à celui décrit plus bas pour les représentations à bases de cellules de Voronoï.

Malgré son succès dans la résolution de problèmes d'optimisation topologique de formes [32, 30, 33], la représentation "bitarray" souffre d'un profonde limitation liée à la dépendance de la complexité de l'algorithme avec celle du maillage associé. En effet, la taille d'un individu (le nombres de bits nécessaires pour décrire un individu) est égale à la taille du maillage. Malheureusement, les résultats théoriques [11] comme les constatations empiriques [21] indiquent que la taille critique de population nécessaire pour atteindre la convergence augmente au moins linéairement avec la taille de chaque individu. De plus, les populations plus nombreuses nécessitent souvent un plus grand nombre de générations pour converger. Il est donc clair que cette approche doit restreindre son domaine d'application à de grossiers maillages bidimensionnels, alors que les ingénieurs ont besoin de fins maillages tridimensionnels!

Ces considérations conduisent à la recherche de représentations plus compactes, dont la complexité ne dépend pas de celle de la discrétisation. Pour obtenir une représentation indépendante de la complexité une ultime étape consiste à la faire évoluer elle-même en l'ajustant par AE.

3 Représentations basées sur les diagrammes de Voronoï

La représentation Voronoï est un premier pas vers les représentations non structurées pour l'optimisation topologique de formes. Elle a été proposée pour la première fois dans [50], et a été utilisée essentiellement pour des problèmes d'identification [52, 51]. Nous rappelons dans cette section la définition de la représentation de Voronoï et nous proposons deux nouvelles représentation qui ont des propriétés analogues.

3.1 Représentation de Voronoï

Diagrammes de Voronoï : Considérons un nombre fini de points $V_0, \ldots, V_N$ (les *sites de Voronoi*) dans un domaine borné donné de $\mathbb{R}^n$ (le domaine de travail). A chaque site V_i on associe l'ensemble de tous les points du domaine de travail pour lesquels le site de

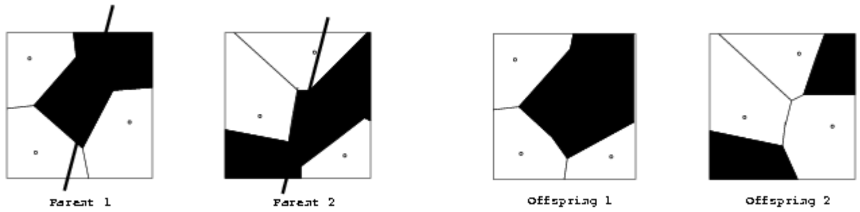


FIG. 2 – *La représentation de Voronoï et ses opérateurs de croisement : une droite aléatoire est tracée dans chaque diagramme et les sites sont échangés d'un coté de cette droite*

Voronoï le plus proche est V_i. On nomme cet ensemble *cellule de Voronoï*. Le *diagramme de Voronoï* est la partition du domaine définie par les cellules de Voronoï. Chaque cellule est un sous-ensemble polyédral du domaine de travail; réciproquement, toute partition d'un domaine de $\mathbb{R}^n$ en sous-ensembles polyédraux est le diagramme de Voronoï d'au moins un ensemble de sites de Voronoï (cf. [48] pour une introduction détaillée aux diagrammes de Voronoï et une présentation générale des algorithmes géométriques).

Le génotype : Considérons maintenant une liste – de longueur variable – de sites de Voronoï, chaque site étant étiqueté 0 ou 1, ainsi que chaque cellule qui lui est associée. Le diagramme de Voronoï correspondant représente une partition du domaine de travail en deux sous-ensembles. Des exemples de représentations de Voronoï sont proposés Figure 2.

Décodage : D'un point de vue pratique, la performance de toutes les structures est évaluée en utilisant un maillage de taille fixe. Une partition décrite par un diagramme de Voronoï peut facilement se projeter sur un maillage donné : un élément appartient à l'une ou l'autre des catégories (matériau ou vide) en fonction du label de la cellule de Voronoï dans laquelle se trouve son centre de gravité.

Initialisation : La procédure d'initialisation consiste en un tirage aléatoire uniforme du nombre de sites de Voronoï (compris entre 1 et un nombre maximum donné par l'utilisateur), des sites de Voronoï dans la structure, et du label booléen vide/matériau.

Opérateurs d'évolution : Les opérateurs d'évolution pour la représentation de Voronoï dépendent du problème :
- L'opérateur de **croisement** échange les sites de Voronoï sur une base géométrique. De ce point de vue, il est similaire à l'opérateur de croisement spécifique décrit dans [31]. La Figure 2 est un exemple d'application de cet opérateur.
- L'opérateur de **mutation** est choisi grâce à une sélection par tirage de roulette, avec des poids définis par l'utilisateur, parmi les opérateurs suivants :
  - la mutation de *déplacement* effectue une mutation Gaussienne sur les coordonnées des sites. Comme dans les stratégies d'évolution [56], la mutation adaptative est utilisée : une déviation standard est associée à chaque coordonnée de chaque site de Voronoï, et elle subit une mutation log-normale avant d'être

utilisée pour la mutation Gaussienne des coordonnées correspondantes.
– la mutation de *label* change aléatoirement l'attribut booléen d'un site.
– les mutations *ajoute* et *supprime* sont des opérateurs spécifiques de longueur variable qui respectivement ajoutent ou suppriment aléatoirement un site dans la liste.

Une analyse par éléments finis est nécessaire pour calculer la performance. Le remaillage étant une source d'erreur numérique supplémentaire dont l'ampleur peut dépasser la différence réelle de comportement mécanique entre deux structures relativement similaires, il est impératif d'utiliser le même maillage pour tous les individus d'une même génération. Néanmoins, la complexité des individus – c-à-d le nombre de sites de Voronoï dans leur représentation – est totalement indépendante du choix de ce maillage. Elle évolue en suivant les principes Darwiniens qui régissent le processus d'évolution.

3.2 La représentation par dipôles

Contrôle de la frontière : Un point important en optimisation topologique est le fin ajustement des frontières de la solution. Une forme optimale ne peut être atteinte en un temps de calcul raisonnable que si l'algorithme est capable de contrôler précisément les frontières des individus d'une population. Malheureusement, la représentation Voronoï ne permet qu'un contrôle indirect de la frontière d'un individu. De plus, par sa structure même, cette représentation rend difficile la modification d'une portion de la frontière sans toucher aux portions adjacentes. La représentation par *dipôles* est une tentative pour palier cette difficulté.

Dipôles : Un dipôle est un ensemble de deux sites de Voronoï, l'un étiqueté 0 et l'autre 1, situés au même point du domaine de calcul, avec une médiane associée repérée par son angle directeur. Un dipôle est donc défini en 2 dimensions d'espace par trois variables réelles : ses coordonnées (x, y) et l'angle θ de sa médiane avec l'axe x. La Figure 3-a est un exemple de dipôle.

Le génotype : Un individu dans la représentation dipolaire est une liste – de longueur variable – de dipôles. Comme dans la représentation Voronoï, le diagramme de Voronoï correspondant donne une partition du domaine de travail en deux sous-ensembles.

Décodage : Comme dans la représentation Voronoï, la performance des individus est évaluée sur un maillage fixe, et la projection sur ce maillage s'effectue comme dans la section 3.1.

Toutefois, comme on peut le voir sur la Figure 3-b, le décodage de deux dipôles adjacents montre que la structure qui en résulte possède deux sortes de frontières : la médiane des dipôles, qui peut être contrôlée par l'algorithme ; et la médiane entre deux dipôles, dont le contrôle est aussi difficile que dans la représentation Voronoï. Il peut même arriver qu'interviennent des situations compliquées, comme celle de la Figure 3-b, où le

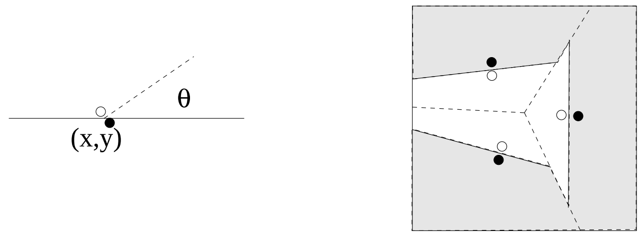


FIG. 3 – *La représentation par dipôles. Un seul dipôle (a) et le diagramme de Voronoï construit à l'aide de trois dipôles (b) : des coins indésirables apparaissent aux croisements des médianes.*

contrôle est encore plus délicat.

Opérateurs d'évolution : Ils sont dérivés de ceux de la représentation de Voronoï : la procédure d'initialisation choisit un nombre de dipôles et initialise leurs coordonnées et angles uniformément dans le domaine de travail $\times[0, 2\pi]$. L'opérateur de croisement échange les dipôles exactement comme les sites dans le cas de la représentation de Voronoï (cf. Figure 2). Les opérateurs de mutation sont également similaires avec une possibilité supplémentaire de mutation Gaussienne sur l'angle d'un dipôle.

3.3 La représentation par barres

Structures en treillis : Pour les problèmes de cantilever, en 2d, il est connu que les formes optimales ont une structure en treillis [44]. L'obtention de telles structures par des diagrammes de Voronoï ou des dipôles requière l'émergence de sous-ensembles de sites ou de dipôles couplés. L'évolution vers des situations de ce type peut demander de très nombreuses générations. L'idée de la représentation par barres de Voronoï est d'aider à l'apparition de structures de ce genre en introduisant des formes élémentaires qui sont déjà des barres.

Les barres de Voronoï : Une barre de Voronoï est définie par quatre variables réelles : ses coordonnées (x, y), l'angle θ de la barre avec l'axe x et sa largeur. La Figure 3.3-a montre un exemple d'une barre de Voronoï.

Le génotype : Un individu dans la représentation par barres de Voronoï est une liste de taille variable de barres. Quand toutes les barres sont simplement vues comme des sites de Voronoï, le diagramme correspondant donne une partition du domaine en polygones convexes. Chaque polygone est alors séparé en deux sous-domaines : la partie centrale – définie par l'angle et la largeur – contient de la matière, et son complémentaire contient du vide (cf. Figure 3.3). Lorsque la largeur est assez grande, la totalité de la cellule est remplie de matière, tandis qu'une valeur nulle de la largeur donne une cellule vide.

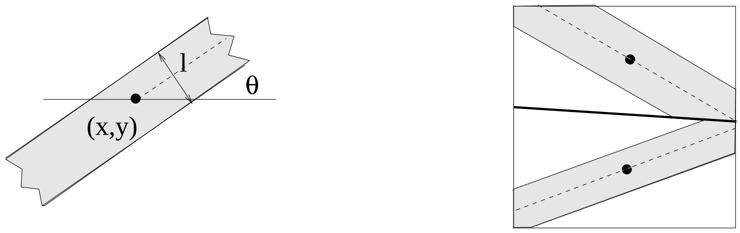


FIG. 4 – *La représentation par barres de Voronoï. Une seule barre (a) et la structure générée par deux barres(b) : le trait plus épais est la frontière entre les deux cellules de Voronoï. Elle ne fait partie de la structure qu'à la jonction entre les deux barres.*

Décodage : Comme pour la représentation Voronoï, la performance des individus est évaluée après projection sur un maillage fixe telle qu'elle est décrite dans la section 3.1 : un élément du maillage est considéré comme plein de matière si et seulement si son centre de gravité est dans la partie pleine d'une barre de Voronoï.

Comme on peut le voir sur la Figure 3.3-b, le décodage de barres de Voronoï adjacentes permet le contrôle direct de presque toute la frontière de la structure, à l'exception des quelques portions limitées à la jonction de deux barres.

Opérateurs d'évolution : Ils sont de nouveau dérivés de ceux de la représentation de Voronoï : la procédure d'initialisation choisit un nombre de barres et initialise leurs coordonnées, angles et largeur uniformément. L'opérateur de croisement échange les barres exactement comme les sites dans le cas de la représentation de Voronoï (cf. Figure 2). Les opérateurs de mutation sont également similaires avec des possibilités supplémentaires de mutation Gaussienne sur l'angle et la largeur de la barre.

4 Résultats numériques avec les représentations à base de diagrammes de Voronoï

Nous présentons ici quelques résultats obtenus avec les représentations à base de diagrammes de Voronoï proposées plus haut. La plupart utilisent la représentations de Voronoï décrite dans la section 3.1 car c'est la première représentation non structurée à avoir été appliquée à des problèmes d'optimisation topologique. Des tests avec différents maillages ont été pratiqués pour s'assurer que la représentation non structurée joue bien le rôle qu'on attend d'elle dans la dépendance de l'algorithme par rapport à la complexité de la discrétisation. Des résultats nouveaux sur un cantilever "allongé" (10×1) est quelques cantilevers tridimensionnels montrent que de telles représentations compactes peuvent permettre de franchir une étape dans la résolution de problèmes d'optimisation topologique par algorithmes évolutionnaires. Enfin, des résultats comparatifs sur le benchmark du cantilever justifient l'introduction des autres représentation de type Voronoï.

4.1 Paramètres numériques pour les algorithmes évolutionnaires

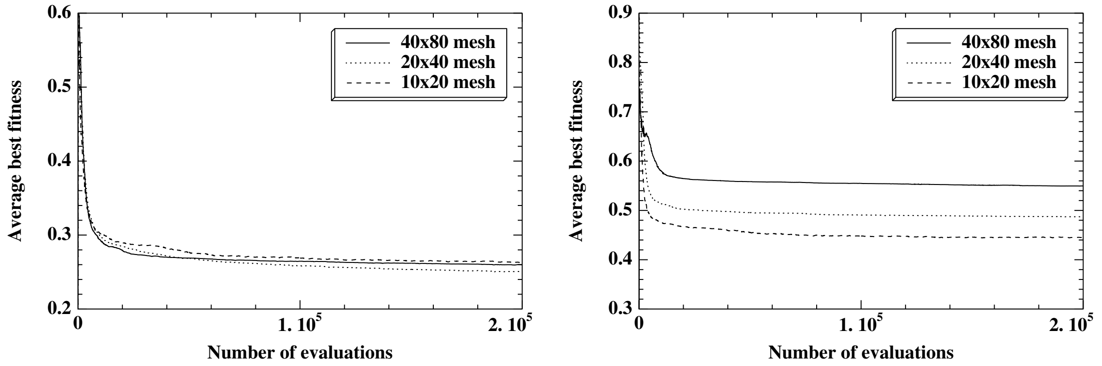
Sauf indication contraire, les expériences numériques présentées dans la suite utilisent les paramètres suivants : évolution standard de type Algorithme Génétique (sélection linéaire basée sur le rang et remplacement de tous les parents par leur descendance) avec des population de 80 individus ; au plus 40 sites de Voronoï (ou dipôles ou barres) par individu ; taux de croisement de 0.6 et taux de mutation de 0.3 par individu ; les poids relatif de chaque type de mutation sont de 0.5 pour la mutation de déplacement, les autres types de mutations se partageant en parts égales les 0.5 restants ; le nombre maximum de génération est de 2000 et le critère d'arrêt de l'algorithme est de 300 itérations successives sans amélioration du meilleur individu ; tous les graphiques sont les résultats de 21 calculs indépendants avec les mêmes paramètres ; les temps CPU sont donnés pour des processeurs Pentium III à 300MHz sous Linux. Par exemple, le coût d'une génération pour le cantilever 1×2 ou 2×1 discrétisé avec 200 éléments finis est de 2s.

4.2 Dépendance par rapport au maillage

Pour vérifier l'indépendance des résultat de la représentation Voronoï par rapport à la complexité du maillage, nous avons testé différentes discrétisations régulières pour le cantilever 2×1, avec les même paramètres mécaniques et d'évolution.

Nous avons regardé les variations du nombre d'évaluations de la performance afin de déterminer si les raffinements de maillages modifiaient la vitesse de convergence, c-à-d le nombre de calculs par éléments finis nécessaires pour atteindre une solution donnée (bien entendu le temps de calcul de chaque évaluation de la performance augmente avec la finesse de la discrétisation sans qu'il soit possible de l'éviter).

La Figure 5 montre l'évolution de la performance du meilleur individu (moyennée sur 21 calculs indépendants) pour des maillages 10×20, 20×40 et 40×80 et deux valeurs différentes de la contrainte sur le déplacement maximum ($D_{lim} = 10$ et $D_{lim} = 20$), le nombre d'individus par génération restant constant (80). Bien que la Figure 5-a montre comme on l'attendait une parfaite indépendance de la vitesse de convergence par rapport à la discrétisation, la Figure 5-b semble contredire cette hypothèse. Toutefois, un examen plus attentif permet d'avancer une explication : la meilleure solution obtenue pour le maillage 10×20 (poids=0.44, déplacement maximal=0.997), une fois projetée sur le maillage plus fin 20×40 donne un poids de 0.43125 et un déplacement maximal de 11.265 ! Ainsi, la grossièreté du maillage est responsable des différences entre les courbes – et comme la différence relative est plus importante pour les structures légères, cela explique en grande partie le second graphique. Notons enfin que, même sur la Figure 5-b, le comportement général de l'AE est identique pour les 3 maillages, à la différence près de la différence entre les solutions finales. De plus, le nombre de sites de Voronoï pour décrire la meilleure solution sur les trois maillages est approximativement le même.



(a) $D_{lim} = 20$ (b) $D_{lim} = 10$

FIG. 5 – *Poids de la meilleure structure au cours de l'évolution (moyenne sur 21 calculs indépendants) pour trois finesses de maillages différentes du cantilever* 1×2.

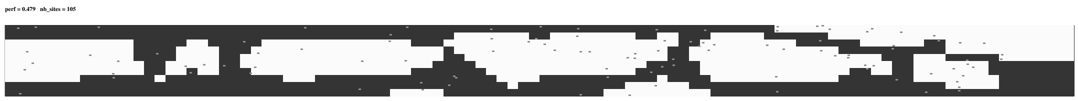


FIG. 6 – *Structure optimale sur le maillage* 100×10 *pour le cantilever* 10×1. $D_{lim} = 12$, *nombre de sites de Voronoï = 105. (représentés par des points en grisé sur la figure), poids = 0.479, temps CPU = 14s/génération.*

4.3 Le cantilever 10×1

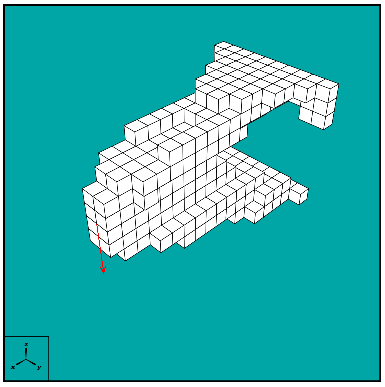
Le problème du cantilever 10×1 (discrétisé par un maillage régulier de 100×10 éléments) est délicat à traiter avec une représentation de type bitarray à cause d'une difficulté supplémentaire par rapport au cas précédent : la plupart des structures générées aléatoirement au cours du processus d'initialisation ne connectent pas le point d'application de la force avec la frontière encastrée. Une procédure d'initialisation particulière est utilisée, dans laquelle le poids moyen des structures aléatoires peut être contrôlé (cf. [29] pour les détails). De plus, le nombre maximal de sites par individus a été augmenté à 120 et les meilleurs résultats sont obtenus pour des populations de 120 individus. La Figure 6 montre un des résultats les plus significatifs pour une contrainte $D_{lim} = 12$.

4.4 Un problème tridimensionnel

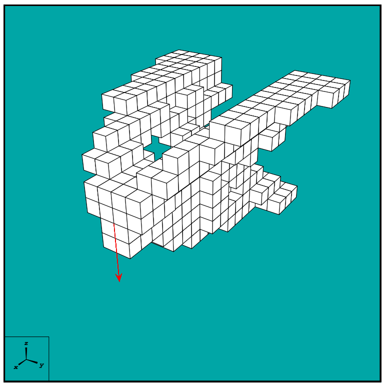
Nous présentons dans cette section les premiers (à notre connaissance) résultats 3d en optimisation topologique de formes par algorithmes évolutionnaires. Ils ont été obtenus en utilisant la représentation par cellules de Voronoï.

Le domaine de travail est un parallélépipède rectangle et le problème présente une symétrie permettant de ne discrétiser que la moitié du domaine par un maillage à $16 \times 7 \times 10$ éléments. Le plan en $x = 0$ (au fond sur la Figure 7) est encastré, et une force verticale $(//z)$ est appliquée au milieu de la face opposée.

Pour ce problème plus complexe que les cas précédents, quelques paramètres doivent



(a) : poids=0.15178, 103 sites



(b) : poids=0.166, 109 sites

FIG. 7 – *Deux résultats pour le cantilever tridimensionnel sur un maillage* $16 \times 7 \times 10$ *sur la moitié du domaine, avec la même contrainte sur le déplacement maximal. Temps CPU = 6mn/génération.*

être modifiés : la taille de la population est de 120 individus et le nombre maximum de site de Voronoï par individu vaut 120.

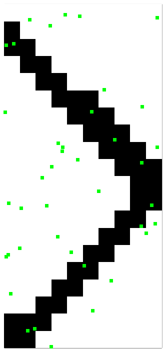
La Figure 7 montre que l'algorithme a été capable de trouver quelques solutions acceptables ...en quelques jours de calcul (les analyses par éléments finis de problèmes 3d sont beaucoup plus coûteuses que pour les cas 2d à nombre de mailles égal). De plus, elle démontre la capacité connue des AE de trouver des solutions quasi-optimales multiples pour un même problème, certaines étant assez originales par rapport à celle obtenue par la méthode par homogénéisation [1].

4.5 Résultats comparatifs pour les représentations à base de diagrammes de Voronoï

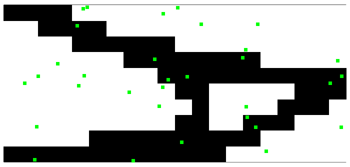
Nous présentons dans cette section les résultats obtenus avec les trois représentations non structurées décrites plus haut sur des benchmarks classiques : nous avons considéré des cantilevers de dimensions 1×2 et 2×1 avec des limites respectives sur le déplacement maximal de 20 et 220. Les paramètres numériques sont ceux indiqués dans la section 4.1.

Les Figures 8, 9 et 10 montrent les meilleures structures obtenues avec les trois représentation (Voronoï, dipôles et barres) et les Figures 11 et 12 montrent des statistiques sur les 21 calculs indépendants pour les deux cas-test.

La première conclusion de ces expériences numériques est que les trois représentations permettent toutes de trouver des solutions aussi bonnes sur les 21 calculs. Toutefois, pour ces deux cas-test, la meilleure représentation est sans conteste la représentation par barres de Voronoï : avec elle, presque toutes les solutions parmi les 21 calculs indépendants sont identiques à celle de la Figure 10, tandis que de nombreuses solutions obtenues par la représentation par dipôles sont plus mauvaises, et encore plus avec la représentation Voronoï. Ces tendances sont visibles sur les comparaisons des Figures 11 et 12. Notons que

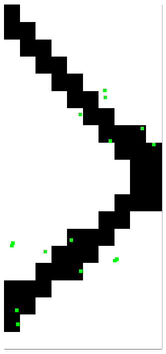


(a) : poids=0.215, 35 sites

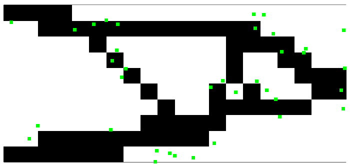


(b) : poids=0.35, 32 sites

FIG. 8 – *Les deux meilleurs individus pour la représentation de Voronoï.*

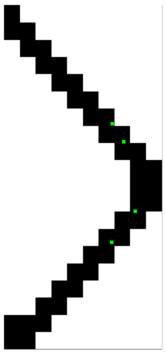


(a) : poids=0.215, 15 dipôles

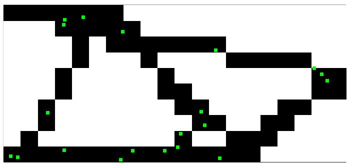


(b) : poids=0.325, 36 dipôles

FIG. 9 – *Les deux meilleurs individus pour la représentation par dipôles.*

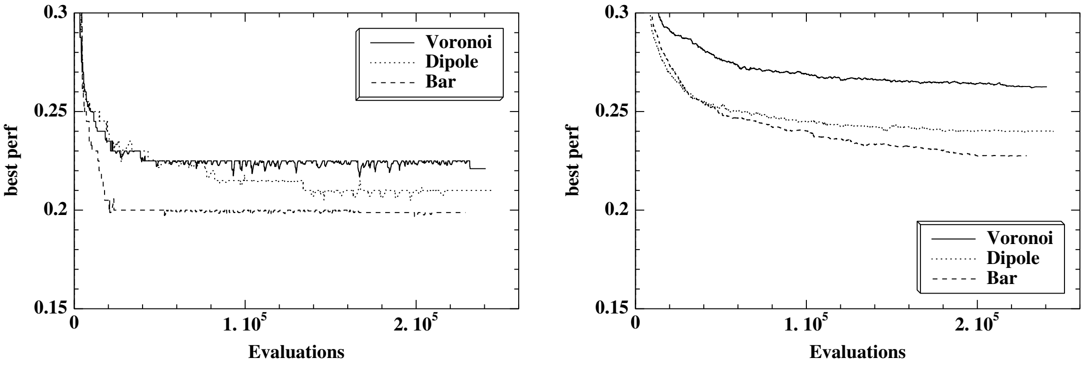


(a) : poids=0.2, 4 barres



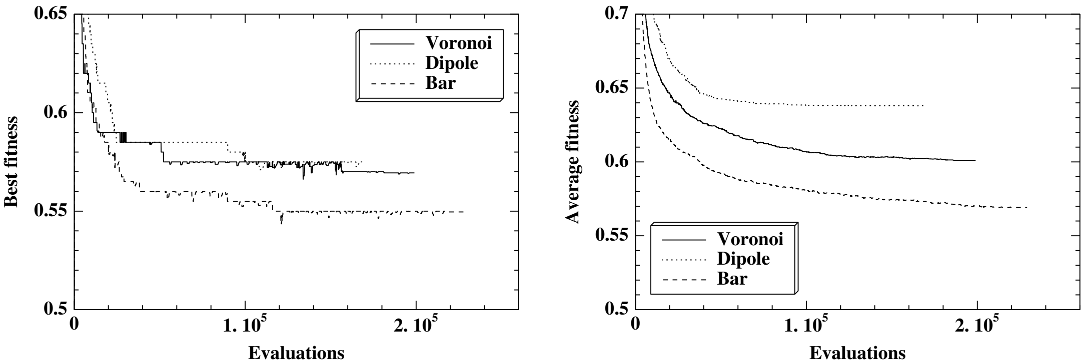
(b) : poids=0.33, 20 barres

FIG. 10 – *Les deux meilleurs individus pour la représentation par barres de Voronoï.*



Meilleur Moyenne

FIG. 11 – *Représentations de Voronoï sur le cantilever* 1×2 *pour Dlim = 20.*



Meilleur Moyenne

FIG. 12 – *Représentations de Voronoï sur le cantilever* 2×1 *pour Dlim = 220.*

les deux autres représentations permettent parfois de trouver des solutions comparables à celles obtenues avec les barres de Voronoï, mais cette dernière s'avère beaucoup plus robuste.

Un autre critère est la complexité des solutions. Les solutions attendues pour ces cas-test sont très simples et la représentation devrait refléter cette simplicité. Sur ce point aussi, la représentation par barres de Voronoï trouve des représentations très compactes comparées aux autres. Un des calculs a même obtenu la structure parfaite formée de deux barres en V pour le cantilever 1×2.

Il semble donc que la complexité supplémentaire introduite dans les "gènes" élémentaires de la représentation par barres de Voronoï soit efficace, au moins sur ces deux benchmarks.

5 Représentations à base de systèmes de fonctions itérées (IFS)

Les représentations à base de diagrammes de Voronoï sont des techniques utilisées pour découpler l'encodage des structures et la discrétisation qui sert au calcul de la performance. Toutefois, les blocs élémentaires servant à construire la structure doivent être définis par le programmeur, et des choix inadéquats peuvent biaiser la recherche dans une mauvaise direction.

La représentation suivante, à base de fractales, est une tentative pour aller plus loin dans la direction morphogénétique : aucune hypothèse n'est faite sur ce que doit être la forme des blocs élémentaires, mais l'espace de recherche pour le génotype est supposé assez riche pour que n'importe quel type de structure puisse apparaître.

Un IFS est un ensemble de transformations contractantes (cf. section suivante pour une définition détaillée). Le principal intérêt d'un tel objet mathématique est qu'il définit de façon unique un ensemble particulier, appelé son "attracteur" (qui est construit simplement grâce à un algorithme itératif). Les attracteurs d'IFS furent utilisés à l'origine pour définir des ensembles fractals (tapis de Sierpinski, fougère de Barnsley, etc.). Mais la principale propriété des IFS est leur capacité à décrire des formes complexes avec un petit nombre de paramètres. Elle est à l'origine du succès de la théorie des IFS dans le domaine de la compression du signal, mais elle a de nombreuses autres applications en analyse d'image ou de signal, en particulier en tant qu'outil de représentation très souple (cf. par exemple [39]).

5.1 Un peu de théorie des IFS

Un IFS $\Omega = \{F, (w_n)_{n=1,..,N}\}$ est un ensemble de N fonctions définies sur un espace métrique complet (F, d). Soit W l'opérateur de Hutchinson défini sur l'espace des sous-ensembles de F :

$$\forall\ K\ \subset\ F,\ \ W(K) = \bigcup_{n \in [0,N]} w_n(K)$$

Si les fonctions w_n sont contractantes (i.e. $\exists s \in [0,1[, d(w(x), w(y)) \leq s.d(x,y)\ \forall (x,y) \in F^2$), l'IFS est appelé *hyperbolique* et il existe un unique ensemble A, appelé l'*attracteur* de l'IFS, tel que $W(A) = A$.

L'unicité de l'attracteur est le résultat du théorème du point fixe pour W, qui est contractante pour la distance de Hausdorff définie par

$$d_H(A,B) = \max[\max_{x \in A}(\min_{y \in B} d(x,y)), \max_{y \in B}(\min_{x \in A} d(x,y))]$$

D'un point de vue algorithmique, il y a deux méthodes connues pour calculer l'attracteur d'un IFS :

- **La méthode stochastique (pile ou face) :** soit x_0 un point fixe de l'une des fonctions w_i. On construit la suite x_n par $x_{n+1} = w_i(x_n)$, i étant choisi

aléatoirement dans $\{1..N\}$. Alors $\bigcup_n x_n$ est une approximation de l'attracteur de Ω (La précision de l'approximation augmente avec n).
- **La méthode déterministe :** à partir de n'importe quel noyau S_0, On construit la suite de sous-ensembles $\{S_n\}$ par $S_{n+1} = W(S_n)$. Lorsque $n \longrightarrow \infty$, S_n est une approximation de l'attracteur de Ω.

5.2 Identification d'IFS par algorithmes évolutionnaires

La première tentative pour faire évoluer des IFS par AE concernait le problème inverse suivant : pour une forme donnée $A \subset F$, trouver un IFS dont l'attracteur est A.

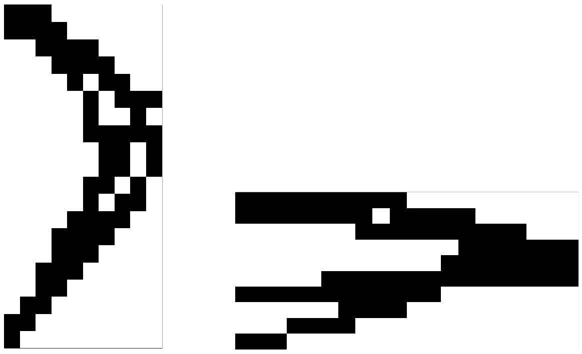
Ce problème peut se reformuler comme un problème d'optimisation : trouver l'IFS dont l'attracteur minimise la distance à l'objectif A. Comme la fonction à optimiser est très complexe, des hypothèses restrictives supplémentaires sont nécessaires. En général, on limite l'espace de recherche à celui des IFS affines, avec un nombre fini de fonctions (cf. [5, 38] pour les premières méthodes numériques). Plus récemment, des solutions basées sur les AE ont été proposées pour les IFS affines [60, 19, 45].

Mais les IFS affines ne sont qu'un petit sous-ensemble de tous les IFS. Dans [41] on s'intéresse à des IFS plus généraux (non affines) appelés *IFS mixtes* en utilisant la programmation génétique (GP) [34] qui permet de faire évoluer n'importe quel type de fonction. Toutefois, alors que la démonstration de la contractivité des fonctions affines est triviale, la contractivité de fonctions générales définies par des arbres de programmation génétique ne peut être montrée que numériquement, *a posteriori*, et pour un coût de calcul très élevé. Cet inconvénient a motivé l'introduction très récente des IFS polaires [14] dans lesquels les fonctions sont définies – toujours en utilisant la GP – sous forme polaire autour de leurs points fixes : une condition simple sur les fonctions ρ assure la contractivité locale autour des points fixes. Cela ne permet pas de vérifier la contractivité globale, mais la proportion de fonctions contractantes dans cette classe de fonctions polaires est bien plus grande que dans les arbres GP généraux, permettant une résolution du problème inverse plus rapide et plus précise.

Malheureusement, au début de ce travail, seuls les programmes de GP pour l'identification des IFS mixtes étaient opérationnels. Les premiers résultats présentés dans les sections suivantes concernant l'utilisation de représentations IFS pour des problèmes d'optimisation topologique ont donc été obtenus en utilisant le programme d'IFS mixtes basé sur la programmation génétique qui est décrit en détails dans [41].

5.3 Représentation par IFS : résultats préliminaires

L'idée de la représentation d'une forme par IFS est évidente : l'attracteur d'un IFS **est** la forme. Le programme utilisé est le même que dans [41]. L'attracteur d'un IFS donné est calculé sur le maillage qui sert à l'analyse par éléments finis et la performance est calculée comme dans la section 2.3. On reprend les cas-test des cantilevers 1×2 et 2×1 de la section 4. La Figure 13 montre les meilleurs résultats obtenus après cinq calculs.



(a) : poids=0.31 (b) : poids=0.43

FIG. 13 – *Résultats pour la représentation par IFS pour les deux cas-test de la section 4.*

La première bonne nouvelle est qu'on obtient des structures raisonnables. De plus, leur allure ressemble plus à un treillis qu'en utilisant la représentation par cellules de Voronoï. Mais on peut entrevoir quelques défauts dans ces toutes premières expérimentations :

- La variance des résultats est très élevée ; certains résultats sont vraiment très mauvais.
- Nous avons utilisé la même stratégie de pénalisation adaptative que pour les représentations à base de diagrammes de Voronoï (cf. section 2.3.2). Pourtant, alors que dans ce cas chaque calcul trouve au moins une solution satisfaisant les contraintes, de nombreux runs utilisant la représentation IFS ne trouvent aucune solution admissible.
- Le temps de calcul du décodage est beaucoup plus important.
- L'influence du raffinement du maillage sur la forme obtenue après décodage de l'IFS n'est pas facile à comprendre, mais il semble que des maillages différents donnent des formes très différentes pour un même IFS, et ce jusqu'à des finesses de maillages très importantes.

6 Conclusion

Nous avons introduit de nouvelles représentations pour le traitement de problèmes d'optimisation topologique de formes par algorithmes évolutionnaires. En partant de la représentation binaire liée à un maillage donné du domaine de calcul, des représentations basées sur la théorie des diagrammes de Voronoï a été proposée, depuis la représentation Voronoï simple jusqu'aux plus complexes représentations par dipôles ou barres de Voronoï. Ces trois représentations sont non structurées, c-à-d qu'un individu est caractérisé par une liste de "gènes" non ordonnée et de longueur variable. La complexité de la structure d'un gène élémentaire augmente de la représentation Voronoï jusqu'à la représentation par barres de Voronoï. Toutefois, ces trois types de représentations permettent d'obtenir des méthodes où la complexité est auto-adaptative, i.e. pour lesquelles la complexité effective des individus évolue à travers l'algorithme.

Nous avons testé ces représentations sur des problèmes-test simple d'optimisation topologique de formes. Les résultats semblent montrer que les trois représentations sont adaptées à la résolution de ce type de problèmes. Elles demandent grossièrement le même temps de calcul pour atteindre des solutions comparables, avec un léger avantage pour la représentation par barres de Voronoï. Pourtant, après examen de la complexité du codage des solutions obtenues, on trouve un avantage net pour la représentation par barres de Voronoï, dont les solutions utilisent moins de "gènes" que les deux autres. Cela explique sans doute le très léger avantage observé pour ce codage en terme de rapport qualité/coût de calcul. En effet, il est plus facile d'ajuster finement la solution lorsque peu de gènes sont en jeu.

Finalement, nous avons présenté la représentation par IFS dans laquelle la structure est définie indirectement comme l'attracteur d'un ensemble de transformations contractantes définies sur le domaine de travail. Une telle représentation ne fait aucune hypothèse *a priori* sur la forme des blocs élémentaires constitutifs d'une solution d'un problème d'optimisation topologique. Cette technique devrait permettre d'atteindre des solutions plus complexes sans avoir à définir de gènes élémentaires spécifiques.

Pourtant, la représentation par IFS n'a pas été capable de trouver d'excellent résultats sur les cas-test simples qui ont été essayés. Bien sûr, il peut être objecté qu'une telle représentation, qui augmente la complexité du processus morphogénétique, ne doit prouver son efficacité que pour des problèmes pour lesquels la solution est complexe. Des études à venir tenteront de confirmer cette hypothèse. Mais il est aussi possible que le manque de feed-back direct de la structure mécanique sur son génotype (l'IFS) empêche des évolutions utiles sur les populations restreintes et de si petits nombres de générations. Des expérimentations sur des machines massivement parallèles avec des populations distribuées de centaines ou de milliers d'individus pourraient répondre à cette question.

Dans l'attente de nouveaux résultats, la représentation par barres de Voronoï semble être un bon choix pour l'optimisation topologique par AE en 2d, réalisant un bon compromis entre la compacité du codage des individus et la facilité de recherche des bonnes solutions. Toutefois, alors que la généralisation des représentations Voronoï et par dipôles à trois dimensions est immédiate (ainsi que celle par IFS), la représentation par barres de Voronoï demande un peu plus d'efforts. De plus, il sera sans doute nécessaire d'introduire des plaques et des barres de différentes sections dans le catalogue des gènes élémentaires.

Références

[1] G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve. Shape optimization by the homogenization method. *Nümerische Mathematik*, 76 :27–68, 1997.

[2] G. Allaire and R. V. Kohn. Optimal design for minimum weight and compliance in plane stress using extremal microstructures. *European Journal of Mechanics, A/Solids*, 12(6) :839–878, 1993.

[3] G. Allaire, Z. Belhachmi and F. Jouve. The homogenization method for topology and shape optimization. Single and multiple loads case. *European Journal of Finite Elements*, 5 :649–672, 1996.

[4] G. Anagnostou, E. Ronquist, and A. Patera. A computational procedure for part design. *Computer Methods in Applied Mechanics and Engineering*, 97 :33–48, 1992.

[5] M. F. Barnsley. *Fractals everywhere.* Academic Press, 1988.

[6] Th. Bäck, D.B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation.* Oxford University Press, 1997.

[7] M. Bendsoe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71 :197–224, 1988.

[8] S. BenHamida and M. Schoenauer. Adaptive penalty for constrained evolutionary optimization. Technical report, Ecole Polytechnique, 2000. Submitted.

[9] O. Bohnenberger, J. Hesser, and R. Männer. Automatic design of truss structures using evolutionary algorithms. In D. B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, volume 1, pages 143–149. IEEE Press, 1995.

[10] J. Cea. Problems of shape optimum design. In E. J. Haug and J. Cea, editors, *Optimization of distributed parameter structures - Vol. II*, volume 50, pages 1005–1088. NATO Series, Series E, 1981.

[11] R. Cerf. An asymptotic theory of genetic algorithms. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, volume 1063 of *LNCS*, pages 37–53. Springer Verlag, 1996.

[12] C. D. Chapman, K. Saitou, and M. J. Jakiela. Genetic algorithms as an approach to configuration and topology design. *Journal of Mechanical Design*, 116 :1005–1012, 1994.

[13] P. G. Ciarlet. *Mathematical Elasticity, Vol I : Three-Dimensional Elasticity.* North-Holland, Amsterdam, 1978.

[14] P. Collet, E. Lutton, F. Raynal, and M. Schoenauer. Polar IFS + Parisian GP = Efficient inverse IFS problem solving. *Genetic Programming and Evolvable Machines*, 2000. To appear.

[15] R. Dorne and J.-K. Hao. An evolutionary approach for frequency assignment in cellular radio networks. In D. B. Fogel, editor, *Proceedings of the Second IEEE International Conference on Evolutionary Computation.* IEEE, IEEE Press, 1995.

[16] R. Dorne and J.-K. Hao. A new genetic local search algorithm for graph coloring. In Th. Bäck, G. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5^{th} Conference on Parallel Problems Solving from Nature*. Springer Verlag, 1998.

[17] A.E. Eiben and Z. Ruttkay. Self-adaptivity for constraint satisfaction : Learning penalty functions. In T. Fukuda, editor, *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, pages 258–261. IEEE Service Center, 1996.

[18] S. Garreau, Ph. Guillaume, and M. Masmoudi. The topological sensitivity for linear isotropic elasticity. In *Proceedings of European Conference on Computational Mechanics*, 1999.

[19] B. Goertzel. Fractal image compression with the genetic algorithm. *Complexity International*, 1, 1994.

[20] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.

[21] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise and the sizing of populations. *Complex Systems*, 6 :333–362, 1992.

[22] D. Grierson and W. Pak. Discrete optimal design using a genetic algorithm. In M. Bendsoe and C. Soares, editors, *Topology Design of Structures*, pages 117–133. NATO Series, 1993.

[23] A. B. Hadj-Alouane and J. C. Bean. A genetic algorithm for the multiple-choice integer program. Technical Report TR 92-50, Department of Industrial and Operations Engineering, The University of Michigan, 1992.

[24] H. Hamda and M. Schoenauer. Adaptive techniques for evolutionary topological optimum design. In I. Parmee, editor, *Evolutionary Design and Manufacture*, pages 123–136, Springer Verlag, 2000.

[25] R. Hinterding, Z. Michalewicz, and A. E. Eiben. Adaptation in evolutionary computation : A survey. In T. Bäck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the Fourth IEEE International Conference on Evolutionary Computation*, pages 65–69. IEEE Press, 1997.

[26] E. Jensen. *Topological Structural Design using Genetic Algorithms.* PhD thesis, Purdue University, November 1992.

[27] J.A. Joines and C.R. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proceedings of the First IEEE International Conference on Evolutionary Computation*, pages 579–584. IEEE Press, 1994.

[28] F. Jouve. *Modélisation mathématique de l'œil en élasticité non-linéaire*, volume RMA 26. Masson Paris, 1993.

[29] L. Kallel and M. Schoenauer. Alternative random initialization in genetic algorithms. In Th. Bäck, editor, *Proceedings of the 7^{th} International Conference on Genetic Algorithms*, pages 268–275. Morgan Kaufmann, 1997.

[30] C. KANE. *Algorithmes génétiques et Optimisation topologique.* PhD thesis, Université de Paris VI, July 1996.

[31] C. KANE AND M. SCHOENAUER. Genetic operators for two-dimensional shape optimization. In J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution*, number 1063 in LNCS. Springer Verlag, Septembre 1995.

[32] C. KANE AND M. SCHOENAUER. Topological optimum design using genetic algorithms. *Control and Cybernetics*, 25(5) :1059–1088, 1996.

[33] C. KANE AND M. SCHOENAUER. Optimisation topologique de formes par algorithmes génétiques. *Revue Française de Mécanique*, 4 :237–246, 1997.

[34] J. R. KOZA. *Genetic Programming : On the Programming of Computers by means of Natural Evolution.* MIT Press, Massachusetts, 1992.

[35] J. R. KOZA AND AL. *Genetic Programming III : Automatic Synthesis of Analog Circuits.* MIT Press, Massachusetts, 1999.

[36] R. LERICHE AND R. T. HAFTKA. Optimization of laminate stacking sequence for buckling load maximization by genetic algorithms. *AIAA Journal*, 31(5) :951–970, May 1993.

[37] R. G. LERICHE, C. KNOPF-LENOIR, AND R. T. HAFTKA. A segregated genetic algorithm for constrained structural optimization. In L. J. Eshelman, editor, *Proceedings of the 6^{th} International Conference on Genetic Algorithms*, pages 558–565, 1995.

[38] J. LÉVY VÉHEL. *Analyse et synthèse d'objets bi-dimensionnels par des méthodes stochastiques.* PhD thesis, Université de Paris Sud, 1988.

[39] J. LÉVY VÉHEL, K. DAOUDI AND E. LUTTON. *Fractal Modeling of Speech Signals.* Fractals, 3(2) :379–382,1994.

[40] C. LIN AND P. HAJELA. Genetic search strategies in large scale optimization. In *Structures, Structural Dynamics, and Materials Conference*, La Jolla, CA, April 1993. AIAA paper #93-1585.

[41] E. LUTTON, J. LÉVY VÉHEL, G. CRETIN, P. GLEVAREC, AND C. ROLL. Mixed IFS : resolution of the inverse problem using genetic programming. *Complex Systems*, 9 :375–398, 1995.

[42] P. MERZ AND B. FREISLEBEN. Genetic local search for the TSP : New results. In T. Bäck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the Fourth IEEE International Conference on Evolutionary Computation*, pages 159–164. IEEE Press, 1997.

[43] Z. MICHALEWICZ AND M. SCHOENAUER. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1) :1–32, 1996.

[44] A. MICHELL. The limits of economy of material in frame-structures. *Phil. Mag.*, 8, 589–597, 1904.

[45] D. J. NETTLETON AND R. GARIGLIANO. Evolutionary algorithms and a fractal inverse problem. *Biosystems*, 33 :221–231, 1994.

[46] S. Obayashi. Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations. In D. Quadraglia, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Sciences*, pages 245–266. John Wiley, 1997.

[47] B. Paechter, R.C. Rankin, A. Cumming, and T. C. Fogarty. Timetabling the classes of an entire university with an evolutionary algorithm. In Th. Bäck, G. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the* 5^{th} *Conference on Parallel Problems Solving from Nature*. Springer Verlag, 1998.

[48] F. P. Preparata and M. I. Shamos. *Computational Geometry : an introduction.* Springer Verlag, 1985.

[49] I. Rechenberg. *Evolutionstrategie : Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution.* Fromman-Hozlboog Verlag, Stuttgart, 1973.

[50] M. Schoenauer. Representations for evolutionary optimization and identification in structural mechanics. In J. Périaux and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Sciences*, pages 443–464. John Wiley, 1995.

[51] M. Schoenauer, A. Ehinger, and B. Braunschweig. Non-parametric identification of geological models. In D.B. Fogel, editor, *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation.* IEEE Press, 1998.

[52] M. Schoenauer, L. Kallel, and F. Jouve. Mechanical inclusions identification by evolutionary computation. *European Journal of Finite Elements*, 5(5-6) :619–648, 1996.

[53] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the* 4^{th} *Conference on Parallel Problems Solving from Nature*, pages 245–254. Springer-Verlag, LNCS 1141, 1996.

[54] M. Schoenauer and Z. Michalewicz. Boundary operators for constrained parameter optimization problems. In Th. Bäck, editor, *Proceedings of the* 7^{th} *International Conference on Genetic Algorithms*, pages 322–329. Morgan Kaufmann, 1997.

[55] M. Schoenauer and Z. Wu. Discrete optimal design of structures by genetic algorithms. In Bernadou and al., editors, *Conférence Nationale sur le Calcul de Structures*, pages 833–842. Hermes, 1993.

[56] H.-P. Schwefel. *Numerical Optimization of Computer Models.* John Wiley & Sons, New-York, 1981. 1995 – 2^{nd} edition.

[57] A. Smith and D. Tate. Genetic optimization using a penalty function. In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, pages 499–503. Morgan Kaufmann, 1993.

[58] P.D. Surry and N.J. Radcliffe. Formal algorithms + formal representations = search strategies. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the* 4^{th} *Conference on Parallel Problems Solving from Nature*, number 1141 in LNCS, pages 366–375. Springer Verlag, 1996.

[59] L. Taïeb and M. Schoenauer. Optimization of direction finders by genetic algorithms. In *Proceedings of GALESIA'95, Genetic Algorithms in Engineering Systems : Innovations and Applications*, pages 23–29. IEE, 1995.

[60] R. Vrscay. Moment and collage methods for the inverse problem of fractal construction with iterated function systems. In *Fractal 90 Conference*, 1990.

(1) CMAP – UMR 7641 – École Polytechnique – 91128 Palaiseau Cedex
(2) Projet Fractales – INRIA – Domaine de Voluceau – BP 105 – 78153 Le Chesnay Cedex
(3) LMS – UMR 7649 – École Polytechnique – 91128 Palaiseau Cedex